

AMENDMENTS TO THE CLAIMS

1-99. (Canceled)

100. (Currently Amended) A computer implemented method comprising:

a source ETL application receiving, from a user, input that selects one or more database objects to be transported from a source database to a target database;

wherein said source database includes source database metadata that describes a structure of database objects of said source database;

wherein said source database metadata identifies a set of tablespaces that store data for the one or more database objects to be transported, and said set of tablespaces is in a format that is understandable by the target database;

wherein said source ETL application includes source ETL metadata, separate from said source database metadata, that describes a structure of said database objects of said source database;

said source ETL application causing generation of a module comprising metadata that describes a structure of said one or more database objects of said source database;

a target ETL application reading said module;

wherein said target database includes target database metadata that describes a structure of database objects of said target database;

wherein said target ETL application includes target ETL metadata, separate from said target database metadata, that describes a structure of said database objects of said target database;

wherein reading said module causes said target ETL application to perform:

modifying said target ETL metadata based on said source ETL metadata read from said module to describe a structure of said one or more database objects of said target database; and

modifying said target database metadata based on said metadata read from said module to describe a structure of said one or more database objects of said source database;

a target database system incorporating a ~~tablespace holding copy of said set of tablespaces that store said data~~ for at least one of said one or more database objects, ~~wherein incorporating said copy of said set of tablespaces includes modifying the target database metadata to define said copy of said set of tablespaces as a set of tablespaces that are used to store said data for at least one of said one or more database objects.~~

101. (Previously Presented) The method of Claim 100, further comprising:
- in response to a failure occurring during the loading of said database objects within said target database, rolling back all changes made during the loading of the database objects to the target database.

102. (Previously Presented) The method of Claim 100, wherein the selected one or more database objects to be transported from a source database to a target database includes a database object that has metadata stored outside of the source database.
103. (Previously Presented) The method of Claim 100, wherein generation of a module includes analyzing the source database metadata for dependencies.
104. (Previously Presented) The method of Claim 103, wherein analyzing the source database metadata for dependencies includes ensuring proper order of loading of the source database metadata into the target database.
105. (Previously Presented) The method of Claim 100, further comprising:
storing said module in one or more files in a source file system.
106. (Previously Presented) The method of Claim 105, further comprising:
said target ETL application performing:
 reading a specification containing information for how to move modules
 from said source file system to a target file system;
 wherein said information comprises a network protocol and the location in
 the source file system of said one or more files; and
 accessing said one or more files in a source file system based on said
 information.

107. (Previously Presented) The method of Claim 106, wherein the network protocol is one of FTP, HTTP, HTTPS, or rsync.

108. (Currently Amended) A computer implemented method comprising:

a source external application receiving, from a user, input that selects one or more database objects, wherein said one or more database objects include an internal database object to be transported from a source database to a target database and an external database object to be transported to a target external application;

wherein said source database includes source database metadata that describes a structure of said internal database object of said source database;

wherein said source database metadata identifies a set of tablespaces that store data for the one or more database objects to be transported, and said set of tablespaces is in a format that is understandable by the target database;

wherein said source external application includes source external application metadata, separate from said source database metadata, that describes said one or more database objects;

said source external application causing generation of a module comprising metadata that describes a structure of said one or more database objects;

a target external application reading said module;

wherein said target database includes target database metadata that describes a structure of said internal database object;

wherein said target external application includes target external metadata, separate from said target database metadata, that describes said one or more database objects; and

wherein said reading said module causes said target external application to perform loading said one or more database objects within said target database and said target external application, wherein loading includes:

modifying said target external metadata to describe said one or more database objects; and

modifying said target database metadata to ~~describe said internal database object~~define a copy of said set of tablespaces as a set of tablespaces that are used to store said data for at least one of said one or more database objects.

109. (Previously Presented) The method of Claim 108, wherein generation of a module includes analyzing the source database metadata for dependencies.

110. (Previously Presented) The method of Claim 109, wherein analyzing the source database metadata for dependencies includes ensuring proper order of loading of the source database metadata into the target database.

111. (Previously Presented) The method of Claim 108, further comprising: storing said module in one or more files in a source file system.

112. (Previously Presented) The method of Claim 111, further comprising:

said target ETL application performing:

reading a specification containing information for how to move modules

from said source file system to a target file system; and

wherein said information comprises a network protocol and the location of

said one or more files; and

accessing said one or more files in a source file system based on said information.

113. (Previously Presented) The method of Claim 112, wherein the network protocol is one of FTP, HTTP, HTTPS, or rsync.

114. (Previously Presented) The method of Claim 108, further comprising:
in response to a failure occurring during the loading of said database objects
within said target database, rolling back all changes made during the
loading of the database objects to the target database.

115. (Previously Presented) The method of Claim 108, wherein said one or more
database objects to be transported from a source database to a target database
includes a database object that has metadata stored outside of the source database.

116. (Currently Amended) A computer-readable volatile or non-volatile storage device
storing one or more sequences of instructions which, when executed by one or
more processors, causes the one or more processors to perform:

a source ETL application receiving, from a user, input that selects one or more database objects to be transported from a source database to a target database;

wherein said source database includes source database metadata that describes a structure of database objects of said source database;

wherein said source database metadata identifies a set of tablespaces that store data for the one or more database objects to be transported, and said set of tablespaces is in a format that is understandable by the target database;

wherein said source ETL application includes source ETL metadata, separate from said source database metadata, that describes a structure of said database objects of said source database;

said source ETL application causing generation of a module comprising metadata that describes the structure of said one or more database objects of said source database;

a target ETL application reading said module;

wherein said target database includes target database metadata that describes a structure of database objects of said target database;

wherein said target ETL application includes target ETL metadata, separate from said target database metadata, that describes a structure of said database objects of said target database;

wherein reading said module causes said target ETL application to perform:

modifying said target ETL metadata based on said source ETL metadata read from said module to describe a structure of said one or more database objects of said target database; and

modifying said target database metadata based on said metadata read from said module to describe the structure of said one or more database objects of said source database;

a target database system incorporating a tablespace holding copy of said set of tablespaces that store said data for at least one of said one or more database objects, wherein incorporating said copy of said set of tablespaces includes modifying the target database metadata to define said copy of said set of tablespaces as a set of tablespaces that are used to store said data for at least one of said one or more database objects.

117. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 116,
- further comprising instructions which, when executed by one or more processors, causes the one or more processors to perform:
- in response to a failure occurring during the loading of said database objects within said target database, rolling back all changes made during the loading of the database objects to the target database.
118. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 116,
- wherein the selected one or more database objects to be transported from a source database to a target database includes a database object that has metadata stored outside of the source database.

119. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 116,
wherein generation of a module includes analyzing the source database metadata for dependencies.
120. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 119,
wherein analyzing the source database metadata for dependencies includes ensuring proper order of loading of the source database metadata into the target database.
121. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 116,
further comprising instructions which, when executed by one or more processors, causes the one or more processors to perform:
storing said module in one or more files in a source file system.
122. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 121,
further comprising instructions which, when executed by one or more processors, causes the one or more processors to perform:
said target ETL application reading a specification containing information for how to move modules from said source file system to a target file system;

wherein said information comprises a network protocol and the location in the source file system of said one or more files; and
said target ETL application accessing said one or more files in a source file system based on said information.

123. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 122,

wherein the network protocol is one of FTP, HTTP, HTTPS, or rsync.

124. (Currently Amended) A computer-readable volatile or non-volatile storage device storing one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform:

a source external application receiving, from a user, input that selects one or more database objects, wherein said one or more database objects include an internal database object to be transported from a source database to a target database and an external database object to be transported to a target external application;

wherein said source database includes source database metadata that describes a structure of said internal database object of said source database;

wherein said source database metadata identifies a set of tablespaces that store data for the one or more database objects to be transported, and said set of tablespaces is in a format that is understandable by the target database;

wherein said source external application includes source external application
 metadata, separate from said source database metadata, that describes said
 one or more database objects;
 said source external application causing generation of a module comprising
 metadata that describes a structure of said one or more database objects;
 a target external application reading said module;
 wherein said target database includes target database metadata that describes a
 structure of said internal database object;
 wherein said target external application includes target external metadata, separate
 from said target database metadata, that describes said one or more
 database objects; and
 wherein said reading said module causes said target external application to
 perform loading said one or more database objects within said target
 database and said target external application, wherein loading includes:
 modifying said target external metadata to describe said one or
 more database objects; and
 modifying said target database metadata to ~~describe said internal~~
~~database object~~define a copy of said set of tablespaces as a
set of tablespaces that are used to store said data for at least
one of said one or more database objects.

125. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 124,
 wherein generation of a module includes analyzing the source database metadata
 for dependencies.

126. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 125,
wherein analyzing the source database metadata for dependencies includes
ensuring proper order of loading of the source database metadata into the
target database.
127. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 124,
further comprising instructions which, when executed by one or more processors,
causes the one or more processors to perform:
storing said module in one or more files in a source file system.
128. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 127,
further comprising instructions which, when executed by one or more processors,
causes the one or more processors to perform:
said target ETL application reading a specification containing information
for how to move modules from said source file system to a target
file system;
wherein said information comprises a network protocol and the
location of said one or more files; and
said target ETL application accessing said one or more files in a source
file system based on said information.

129. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 128,
wherein the network protocol is one of FTP, HTTP, HTTPS, or rsync.
130. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 124,
further comprising instructions which, when executed by one or more processors,
causes the one or more processors to perform:
in response to a failure occurring during the loading of said database
objects within said target database, rolling back all changes made
during the loading of the database objects to the target database.
131. (Previously Presented) The computer-readable volatile or non-volatile storage device of Claim 124, wherein said one or more database objects to be transported from a source database to a target database includes a database object that has metadata stored outside of the source database.